Perspective and Geometry Approaches to Mouse Cursor Control in Spatial Augmented Reality

Daekun Kim
d332kim@uwaterloo.ca
Cheriton School of Computer Science, University of Waterloo
Canada

Nikhita Joshi
nvjoshi@uwaterloo.ca
Cheriton School of Computer Science, University of Waterloo
Canada

Daniel Vogel
dvogel@uwaterloo.ca
Cheriton School of Computer Science, University of Waterloo
Canada

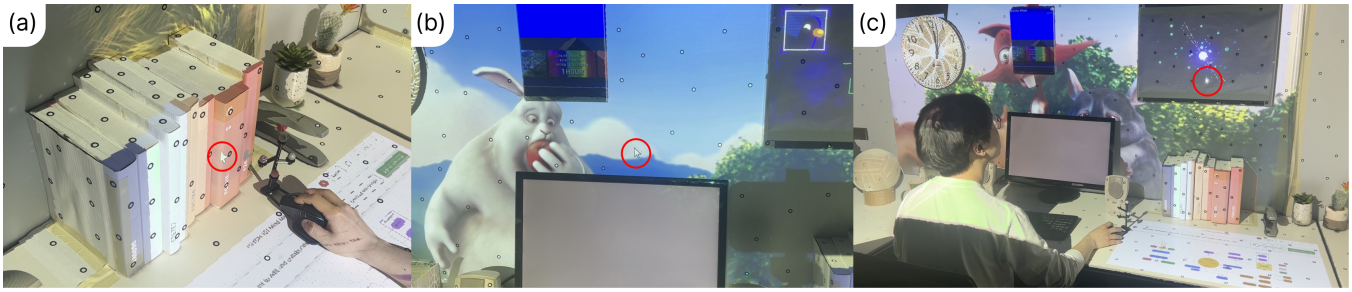

Figure 1: Interacting with SAR contents using a mouse: (a) configuring a projection mapping decoration on a bookshelf; (b) moving the cursor outside the monitor onto a wall; (c) interacting with physical objects like a wall calendar.

ABSTRACT

Spatial augmented reality (SAR) can extend desktop computing out of the monitor and into our surroundings, but extending the standard style of mouse input is challenging due to real-world geometry irregularity, gaps, and occlusion. We identify two general approaches for controlling a mouse cursor in SAR: perspective-based approaches based on raycasting, such as Nacenta et. al's Perspective Cursor, and geometry-based approaches that closely associate cursor movement with surface topology. For the latter, we introduce Everywhere Cursor, a geometry-based approach for indirect mouse cursor control for complex 3D surface geometry in SAR. A controlled experiment compares approaches. Results show the geometry-based Everywhere Cursor improves accuracy and precision by 29% to 60% on average in a tracing task, but when traversing long distances, the perspective-based Perspective Cursor and Raycasting techniques are 22% to 49% faster, albeit with 4% to 10% higher error rates.

CCS CONCEPTS

• **Human-centered computing** → **Pointing**; **Empirical studies in HCI**.

KEYWORDS

Interaction Techniques; Empirical Study

ACM Reference Format:

Daekun Kim, Nikhita Joshi, and Daniel Vogel. 2023. Perspective and Geometry Approaches to Mouse Cursor Control in Spatial Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3544548.3580849

CHI '23, April 23–28, 2023, Hamburg, Germany

1 INTRODUCTION

Spatial Augmented Reality (SAR) is a method to render digital content directly onto surfaces in a real physical 3D environment, typically using projection mapping [24, 25]. It can cover surfaces and objects with illusionary textures and virtual 3D objects, enabling applications like gaming and teleportation (*e.g.*, [11, 20]). But SAR can also be used in a more subtle and integrated way, to selectively augment real surfaces and objects with 2D "surface-mapped" digital information [8]. Essentially, every surface becomes a potential computer display without the need for special AR glasses.

A compelling use case for surface-mapped SAR is to extend the display space for traditional desktop computing, making it pervasive and integrated into the surrounding environment. Potential use cases include: increased screen space [13]; spatial anchoring of information [21]; interactive ambient displays [9]; and sharing of desktop information between digital devices [26, 27]. Imagine using a nearby plant pot as a cylindrical ambient notification display or storing favourite file folders on the spines of books on a shelf.

A key decision when extending desktop computing into a surface-mapped SAR environment is what pointing method to use. Direct touch would only be reasonable for nearby targets and can be less precise [28]. Mid-air raycasting using hand tracking or 3D controllers has issues like tremors, low input precision on distant targets, and 3D surface occlusion [3, 30]. However, using a standard mouse for relative indirect cursor control reduces limitations in terms of target distance or surface visibility, given that its behaviour is consistent with what is familiar in desktop computing. Moreover, unlike touch and raycasting, using the mouse to interact with the surrounding SAR environment would make desktop and SAR input feel integrated due to most computer users' familiarity with it.

Previous work has suggested mouse-to-cursor mappings for environments resembling limited forms of SAR. Early examples use indirect mouse pointing [16, 27] across simple combinations of a few tabletops and walls. Later techniques support more diverse configurations of multiple digital displays [18, 32] and across arbitrary surfaces in small SAR setups [7, 17]. These all place the cursor using some variation of raycasting controlled by the mouse, which we refer to as a *perspective-based approach*. This means they may share disadvantages with standard raycasting, and changing the fundamental mouse pointing paradigm could make interaction inconsistent between a traditional desktop and SAR.

An alternative is to use a *geometry-based approach*, where the cursor moves along the geometry of a display surface exclusively using corresponding relative X and Y mouse movements. To explore this paradigm, we introduce *Everywhere Cursor*, a geometry-based mouse interaction technique for SAR that closely resembles indirect mouse cursor control in conventional desktop computing. It uses a novel combination of near-surface cursor projection and surface traversal with careful filtering and cursor orientation rules. A geometry-based approach directly supports the unique limitation of surface-mapped SAR, where all content (including the cursor) must be rendered on existing environment surfaces.

We compare geometry-based movement enabled by our technique with SAR adaptations of Perspective Cursor [18] and standard raycasting, which represent baseline perspective-based techniques.[1] Our results show that our geometry-based technique was comparable to Perspective Cursor for short-distance target selection tasks. It achieves 29% to 60% improvement in precision and accuracy for more demanding tracing tasks, especially over oblique, irregular, and spherical surfaces. For long-distance movements, perspective-based techniques are 22% to 49% faster over surfaces with complex geometry, but with 4% to 10% higher error rate. We discuss applications of SAR mouse interaction to highlight how each approach is best used, and we outline a future design for a hybrid technique that combines Everywhere Cursor with Perspective Cursor.

We make three main contributions: (1) a formal definition of perspective-based and geometry-based cursor control for expanded desktop environments; (2) the first geometry-based mouse cursor technique specifically designed for SAR; and (3) the first controlled comparison between perspective-based and geometry-based cursor control for both selection and tracing in a complex SAR environment.

2 BACKGROUND AND RELATED WORK

We first provide an overview of work that motivates the expansion of desktops to SAR as a pervasive computing platform. Then, we describe relevant mouse techniques for SAR and multi-display environments that can enable mouse interaction on real physical surfaces. While mouse techniques for VR exist (*e.g.*, [3, 33]), we limit our investigation to non-VR techniques where the cursor must be rendered on existing environment geometry.

[1]This paper significantly expands on a poster by Kim and Vogel [14]: the geometry-based technique uses a new movement algorithm; there is now a controlled experiment; and the contributions and content are significantly expanded into a comparison of perspective-based and geometry-based approaches.

2.1 Large-scale Personal Computing

Previous work that extends personal computing to non-desktop display environments inspires and motivates our approach. Augmented Surfaces [27] integrate digital devices (*e.g.*, laptops and PDAs) and projection surfaces, enabling common mouse interactions like drag-and-drop. While they demonstrate a smooth cursor movement between laptop monitor and projection areas, their system only operates with planar projections, such as tabletops and walls, which limits its generalizability. Nakashima et al. [19] introduce a tabletop multi-user mouse technique that focuses on the importance of aligning the cursor X- and Y-axes with user orientation on a single tabletop surface. Other systems extend desktops into multiple displays and projection areas [13, 16, 22] but do not discuss the specifics of cross-display mouse interaction or mainly use direct touch as the input method.

Raycasting is another popular interaction technique on large-scale computing platforms. The technique employs a *perspective-based* cursor movement, a cursor mapping method using raycast that is fully independent of surface geometry. Vogel et al. [30] found that direct raycasting is more error-prone than relative pointing in freehand pointing. Jota et al. [12] found that techniques with low visual parallax are best for tracing tasks. However, raycasting is heavily affected by hand tremors, has lower precision as target depth increases, and suffers from frequent surface occlusion. More critically, cursor mapping using raycast solely relies on the controller's perspective. For instance, given two surfaces oriented at oblique angles or placed at varying depths, the cursor can change its linear speed even when constant controller velocity is maintained. It may even skip parts of the surfaces occluded from the controller's point of view. In desktop environments, the cursor moves along the plane of the monitor without skipping any part of the surface. Additionally, raycasting is an absolute pointing method, which behaves distinctly from desktop mouse pointing. These discrepancies in movement behaviour limit the interaction consistency of raycasting with a traditional desktop.

2.2 Mouse Interaction in SAR

Expanding desktop interfaces and information to SAR suggests also extending relative mouse interaction to arbitrary surfaces. Viewpoint Cursor [17] maps mouse movement to a point in the user's view plane by raycasting from the head position to find the cursor position on a 3D surface, but it was not evaluated. Gervais et al. [6] formally compared a mouse technique with the same mechanism as Viewpoint Cursor with a standard desktop mouse. Their focus was exploring differences between physical and virtual 3D pointing, and they found both techniques comparable in terms of time and accuracy when selecting targets. However, the SAR environment in the evaluation was a small desk with the targets placed on the front face of a single cube. This simple setup restricts the applicability of their findings to arbitrary environment geometry. Tangible Viewport [7] applies this same mouse technique to SAR content design but restricts the interaction domain to objects placed in front of a monitor. There was an exploratory study focused on the user experience and potential use cases, but no controlled experiment with a baseline. These techniques share disadvantages with raycasting, such as low

Table 1: Comparison to work which proposed and formally compared mouse-based cursor control methods for SAR or MDE.

Paper	Test Environment	Proposed Technique(s)	Device	Baselines	Type of tasks
Gervais et al. [6]	SAR, small desk with cube and screen	Perspective-based (ray from head controlled by mouse)	Mouse and head	Traditional screen	Selection
Ubiquitous Cursor [32]	MDE, all planar	Perspective-based (ray from fixed point on ceiling controlled by mouse)	Mouse and ceiling mirror	Perspective Cursor; stitched displays	Selection
Perspective Cursor [18]	MDE, all planar	Perspective-based (ray from head controlled by mouse)	Mouse and head	Raycasting; stitched displays	Selection
Our work	SAR, large office space with diverse objects including walls	Geometry-based (cursor movement mapped to surfaces)	Mouse	Raycasting; Perspective Cursor	Selection; Tracing

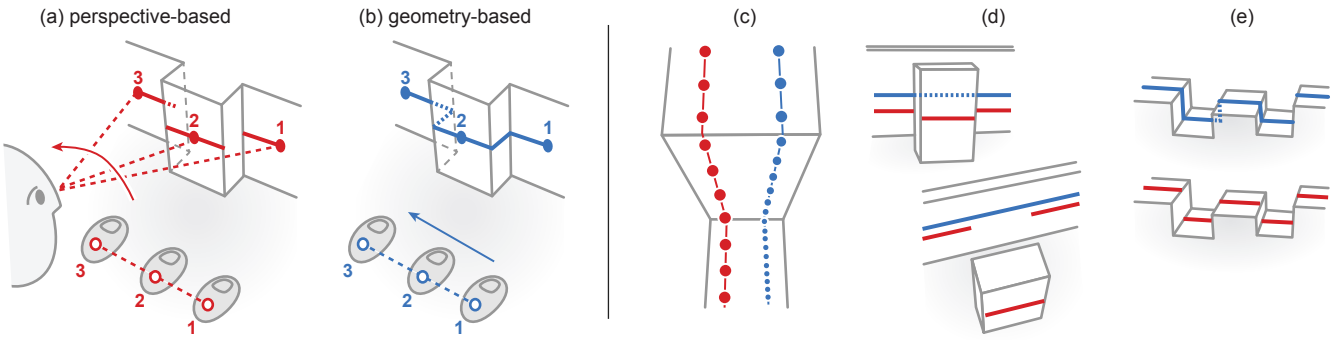

Figure 2: Cursor control approaches: (a) *perspective-based*, mouse movement controls the angle of ray that intersects the nearest surface to place cursor; (b) *geometry-based*, the cursor moves along surface relative to the direction of mouse movement; (c) cursor speed is constant relative to user's view for perspective-based, but relative to traversed surface distance in geometry-based; (d) cursor always appears on the closest surface with perspective-based, but can travel into occluded areas with geometry-based; (e) cursor skips portions of detailed geometry with perspective-based, but can traverse all parts of surface geometry with geometry-based.

input precision on distant targets and sensitivity to surface occlusion. Moreover, the control-display gain (or CD gain) varies with target depth, which is not the case for traditional desktop cursor mapping. This may introduce some interaction inconsistency with respect to typical desktop mouse control.

2.3 Multi-Display Environment Mouse Input

While distinctly unique from projection-mapped SAR, multi-display environments can also provide a similar experience by surrounding the user with digital content on dedicated conventional displays. PointRight [10] and Deskotheque [23] enable mouse cursor movement across separated planar displays that are not necessarily coplanar using stitching and warping, but they were not evaluated. Ubiquitous Cursor [32] is a mouse technique that raycasts through a ceiling-centred hemisphere point to find the cursor position. It visualizes the cursor even on non-monitor areas using projection. Perspective Cursor [18] moves the cursor along a virtual sphere centred around the user's head using raycasting. The experimental evaluation of this technique shows that it is faster and more accurate than standard controller raycasting on selection tasks. Perspective Cursor is essentially a refined version of the SAR and MDE raycasting mouse techniques discussed above, the method is replicable, and it was rigorously tested. For these reasons, we adapt Perspective Cursor to SAR as a canonical example of this approach.

Table 1 summarizes the three most related works that proposed a cursor control technique and conducted a formal experiment. Our work is the first to propose a geometry-based technique, evaluate a tracing task, and compare techniques in a large complex SAR environment.

3 APPROACHES FOR SAR CURSOR CONTROL

There are two general approaches for controlling cursor movement in SAR, perspective-based and geometry-based (Figure 2).

With a *perspective-based* approach, mouse movement is mapped to the azimuth (horizontal) and zenith (vertical) angles of a ray cast from an origin point in the environment. The intersection of this ray with the nearest display surface determines the cursor position. A simple perspective-based approach is to set the origin point to the mouse position (essentially the same as raycasting using a VR controller) or to use a fixed position like Ubiquitous Cursor [32] which casts the ray from a hemisphere in the ceiling. Several works use the user's head as a single ray origin [6, 17], with Perspective Cursor [18] expanding this so the ray is cast from the surface of a sphere centred at the user's head. Conceptually, in all these perspective-based approaches, the cursor position is closely associated with the user's view of the SAR environment.

With a *geometry-based* approach, the cursor moves along the geometry of the display surface exclusively using corresponding

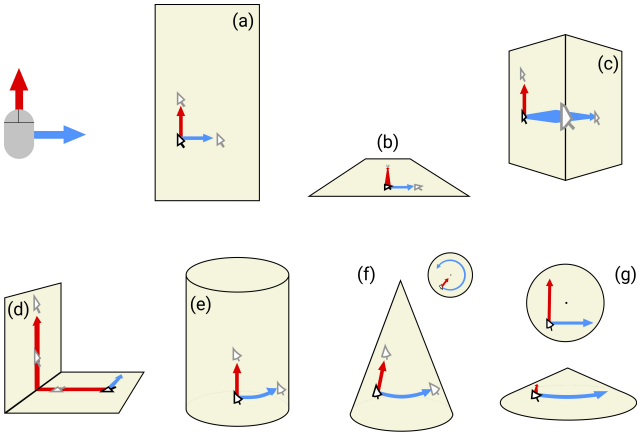

Figure 3: Examples of diverse surface geometry in SAR with ideal cursor X and Y travel path: (a) planar wall, (b) planar tabletop, (c) convex corner, (d) concave corner, (e) cylinder, (f) steep cone, (g) flat cone.

relative X and Y mouse movements. This is precisely how a conventional desktop cursor works on a planar monitor. Conceptually, the cursor position is more closely associated with the geometry and topology of the SAR environment.

Each approach has theoretical advantages and challenges. A perspective-based approach enforces a clear user-centred frame-of-reference which keeps the cursor in view and simplifies cursor movement across complex surface geometry. However, the fundamental dependence on raycasting introduces limitations, such as lower precision on distant targets and difficulty handling surface occlusion. A geometry-based approach is arguably more "desktop-native", since desktop GUIs are designed on the 2D Euclidean coordinate space with linear measures compared to a spherical coordinate system with angular measures implicitly used in perspective-based approaches. However, generalizing geometry-based cursor control to non-planar, diverse, and complex surfaces in a SAR environment is difficult. Below, we identify four main design challenges.

Cursor Orientation and Movement Direction. On a desktop computer, the cursor points in a consistent "upright" orientation as it moves in a predictable up-down and left-right motion along the plane of the monitor. Since perspective-based techniques are relative to the user's view, the cursor movement direction is well-defined, regardless of surface geometry. However, with both perspective- and geometry-based approaches, the rules for rendering the correct cursor orientation relative to a surface are not immediately obvious. Consider the examples in Figure 3: in general, the cursor should be pointing along the global up-axis on vertical surfaces and along the user's forward axis on horizontal surfaces. The user's forward axis is already available in a perspective-based technique like Perspective Cursor where the user's head must be tracked. Otherwise, an approximation of the user's position could be based on mouse position or typical user locations.

Smooth Cursor Trajectory. Given a constant mouse velocity, the perceived cursor velocity should also appear constant along the movement trajectory (constant "CD-Gain" [5]). A perspective-based approach will result in the cursor travelling at different speeds depending on the surface angle relative to the ray (Figure 2c). This could make cursor movement less controllable or appear less connected with the surface. By definition, a geometry-based approach will consistently map mouse velocity to actual cursor velocity along a surface. However, if there are small bumps or scan errors in the surface geometry (below the scale of the rendered cursor), a geometry-based approach using strict surface traversal will result in jittery cursor behaviour.

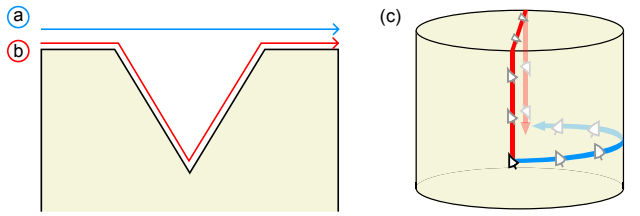

Figure 4: Potential cursor trajectories: (a) skipping across a crevice; (b) moving along a concave surface; (c) natural transition of cursor orientation in two paths appearing to contradict.

Balancing Speed and Interaction with Fine Detail. Due to real world object arrangements and scanning quality, a SAR environment often contains empty regions ("gaps"), deep deviations ("crevices"), and sudden changes in surface depth ("cliffs"). This introduces challenges for mapping cursor movement. Imagine a bookshelf with books separated by crevices: the user may wish to skip across crevices to select books by the spine, or they may want to enter a crevice between two books to access detailed content on a book cover. A simplified example of these two paths with a V-shaped crevice is illustrated in Figure 4a,b. Cliffs have a similar choice of two paths, but gaps necessitate some method of skipping over. A perspective-based approach essentially skips over gaps, crevices, and cliffs, but entering a crevice or traversing a cliff face is difficult due to occlusion and varying cursor trajectory. A strict geometry-based approach will always enter a crevice, traverse the cliff face, or fall into a gap, unless the user chooses a longer cursor path over nearby surfaces to bypass these features.

Frequent Occlusions from Complex Geometry. Some arrangements of objects and surfaces could temporarily hide parts of the environment from the user. While fully-occluded areas are less likely to be used for interaction, partially-occluded surface patches could be. Imagine a web browser on a tabletop with a dish occluding part of the web page or content placed on a row of books where variations in book height hides small surface patches. Interacting with such areas, or at least gracefully travelling through them, is crucial for universal direct manipulation access throughout the environment. Achieving this with a perspective-based approach is challenging since raycasting only places the cursor on non-occluded surfaces relative to the ray origin (Figure 2c). A geometry-based approach allows cursor movement onto or through occluded surfaces, but it is still challenging to maintain a consistent cursor orientation with respect to the first challenge above (Figure 4c).

4 EVERYWHERE CURSOR: GEOMETRY-BASED TECHNIQUE

In this section, we describe such a geometry-based cursor control technique for SAR, called *Everywhere Cursor*. It addresses the cursor control design challenges just discussed using a novel combination of near-surface cursor projection and surface traversal with careful filtering and cursor orientation rules.

4.1 Orthographic Cursor Projection

For cursor visualization, we use a small virtual orthogonal projector, called the *cursor projector*, that floats throughout the 3D model of the environment. It renders a cursor 1.5cm in width and 3cm in height when cast perpendicular to the surface (Figure 5). From the top left corner corresponding to the tip of the cursor icon, a raycast determines a single point in the environment to send mouse events like clicking, dragging, and scrolling to a UI system like Unity. This means any standard UI component is compatible.

We justify the decision to use a cursor projector as follows. In SAR, any content without a direct mapping onto the environment geometry results in a blurry image [11]. A common method of 2D content placement is texture mapping. A naive approach would render and move the cursor directly on the 2D texture map of the SAR environment. However, this presents two issues. First, when the cursor moves across two surfaces that share an edge, an automatically-generated texture map typically has discontinuous and isolated regions (Figure 6a). In general, there is no straightforward way of finding a path in texture space to move the cursor across two isolated texture patches. Even tediously mapping shared transition edges for every pair of coinciding regions leaves the challenge of computing the "upright" orientation in texture coordinates. Secondly, when the cursor visualization straddles two or more surfaces, such as around corners, resolving which portion of the cursor image should go to which texture map region is difficult in texture space (Figure 6b). Using a virtual cursor projector allows cursor placement to be based on the 3D model of the environment itself rather than on its texture map, which removes these complexities.

4.2 Cursor Projection on Irregular Surfaces

We employ a set of six short raycasts around the cursor, referred to as *legs*, to sample nearby surface normals and orient the cursor projector to the surrounding local geometry (Figure 5). Intuitively, the cursor uses legs (like a bug) to orient itself approximately perpendicular to nearby surfaces. Note the legs are aligned with the direction of movement and are not visible to the user. Using these legs to follow and adjust to surface changes creates a geometry-based cursor movement. Because the surrounding surfaces have varying normals, we calculate their average $\hat{n}_{avg}$ as an approximation of the perpendicular direction with respect to nearby surfaces. The legs pointing downward detect the floor, those pointing horizontally above the floor detect any surrounding walls, and those below the floor detect convex corners ahead of the cursor. The cursor projector's up vector is set to $\hat{n}_{avg}$. The use of multiple surface normal samples prevents sudden and erratic changes in the orientation of the cursor projector, which allows the cursor trajectory to remain smooth and robust against minute geometric fluctuations. With

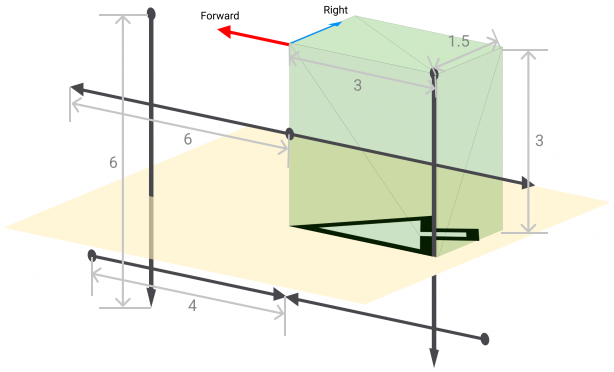

Figure 5: Structure of the orthographic cursor projector (shown as a green box) and its six legs (shown as dark arrows) along the plane of movement (measurements in centimetres).

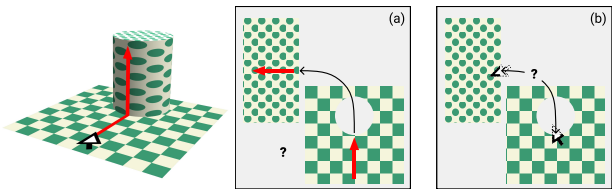

Figure 6: Problems with placing a cursor on a texture map: (a) movement across surfaces that are isolated in the texture space; (b) splitting cursor image across two regions.

geometry-based movement, the technique is also robust against occlusion, as the cursor movement only concerns the local geometry near the cursor rather than surface visibility from a given location (*e.g.*, from the user's head [18] or a device used to raycast).

The 1€ Filter is applied to the leg-adjusted orientation at every frame to prevent jitter and hysteresis [5]. While we did not perform systematic tuning of the filter parameters, we found $f_{c_{min}} = 1, \beta = 1$ yield minimal jitter and lag through trial and error.

4.3 Upright Cursor Orientation

We address the cursor orientation challenge by first determining whether the nearby surfaces are horizontal or vertical and setting the appropriate forward direction. We define the cursor surface as *horizontal* if the angle between $\hat{n}_{avg}$ and the global up-vector is less than 45°, and *vertical* if otherwise. In the horizontal case, the system projects the mouse's forward vector (a proxy for the user's forward direction) onto a plane with normal $\hat{n}_{avg}$. Then, it sets the cursor projector's forward direction to the projection's resulting vector. The vertical case is identical, except the global up vector is used in place of the user's forward vector. This way, the cursor behaves exactly like in a desktop computer when on a wall or tabletop, but it also makes sense on arbitrary surfaces since orientation is well-defined for any type of geometry. A hard angle threshold can cause erratic orientation changes when moving across certain types of curved or irregular surfaces. To address this, the cursor only rotates to the locally determined upright orientation when idle for 0.5s. This means the cursor orientation remains consistent

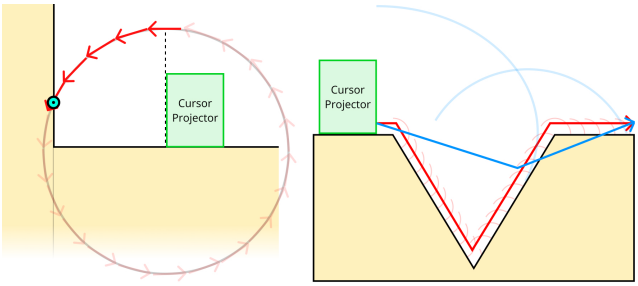

Figure 7: The behaviour of circular raycast: (a) identification of next cursor position using a series of small linear raycasts; (b) faster mouse movement (blue trajectory) translates to lower sensitivity to geometry compared to slower mouse movement (red trajectory).

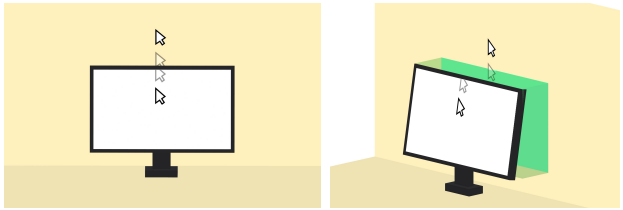

Figure 8: Cursor travelling between monitor and SAR environment, viewed from (a) the user's perspective; (b) the side (hidden region highlighted, not shown in actual system).

during a ballistic movement, but when the user pauses to make a corrective movement, they register where the cursor is, along with its orientation, to plan the next movement.

4.4 Formation of Cursor Trajectory

We use a "*circular raycast*" to find the next cursor position. The system maps the X and Y mouse deltas to cursor projector movement along its local right and forward vectors. Then, it aligns the circular raycast along the direction of movement, with a radius equal to mouse movement distance as determined by system pointer acceleration and CD gain (Figure 7a). The circular raycast consists of a series of linear raycasts, which tangents to the circle. Each ray is tested for surface intersection, starting from the top of the circle and rotating forward until a surface is detected. The intersection is the next cursor position, upon which the cursor projector orientation is adjusted and placed as described above. Faster mouse movement translates to a larger circular raycast radius, creating a speed-based sensitivity to geometry details: moving the mouse faster reduces the influence of obstructive regions like gaps, crevices, and cliffs because rays tangent to a large diameter circle will reach beyond them (Figure 7b).

4.5 Desktop Monitor Transitioning

We expect a monitor to remain a primary surface for interaction, so smooth transitions out of and into the monitor are crucial for effectively integrating SAR into desktop computing. A typical monitor is largely a geometrically isolated surface except for a narrow stand,

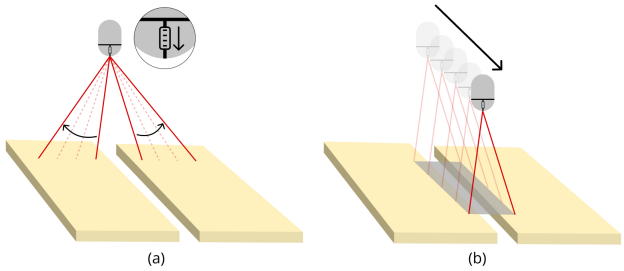

Figure 9: The mechanism for user-guided gap-filling: (a) scroll wheel controls its width; (b) left-clicking and dragging creates a bridge connecting two surfaces.

so an intuitive and easy-to-navigate geometric path between the desktop and SAR does not exist. We address this as a special case, where the cursor position "jumps" to the monitor when entering the hidden region of the wall behind the monitor when viewed orthogonal to the display (Figure 8). Similarly, moving the cursor outside the edge of the display will jump it back to the corresponding position on the wall in SAR.

4.6 User-Guided Gap-filling

As noted above, a SAR environment contains many gaps due to scan errors or object separation. The circular raycast cursor trajectory method is robust enough to cross small gaps with reasonable mouse speed. However, at slow speeds, the cursor can become lost, and this problem is more pronounced with wide gaps. Automated hole-filling algorithms designed for arbitrary 3D meshes can fix small holes and gaps, but we found a SAR environment has gaps that these algorithms do not fill, such as spaces between isolated planes (like side-by-side tables). To address this, we created a *gap-filling tool* where the user interactively creates ad hoc virtual bridges using the mouse (Figure 9). Pressing the mouse's "thumb button" invokes the tool where the virtual SAR geometry is rendered on the monitor as though the mouse was a camera. Then, left-clicking and dragging "paints" bridges between gaps with the bridge width adjusted using the scroll wheel. The gap-filling bridges are tessellated quad meshes the intersect with existing surfaces spanned by the tool. Once the user creates a bridge, it is permanent, meaning this only needs to be done once for each large gap they wish to bridge. Note that bridges are not rendered in the actual SAR space, but the mouse behaviour described above uses the bridges as additional geometry.

4.7 Raycast "Clutching" as a Reset

Despite our best efforts to enable continuous access throughout the room, there are circumstances where the cursor could become "stuck" on a surface region or lost between two disconnected surfaces. To recover from these situations, we implemented a *raycast clutch* as a fallback mechanism to reset the cursor position. Lifting the mouse and tilting it more than 45° to one side positions the cursor at the first intersection of a ray extending out of the front of the mouse. Activation by tilting disambiguates from standard mouse clutching. When raycasting, the 1€ Filter is applied to the mouse position and orientation to prevent jitter, with filter parameters of $f_{c_{min}} = 0.1, \beta = 100$. Tilting the mouse back to its normal

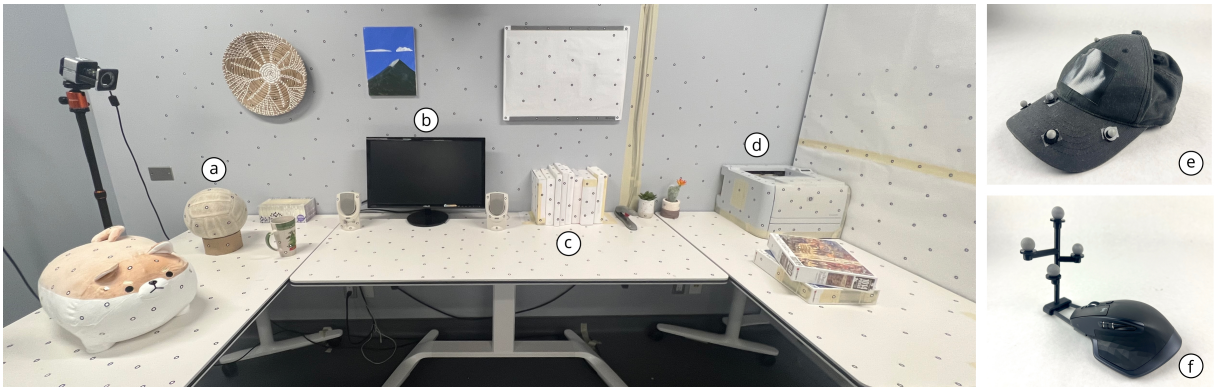

Figure 10: Setup of the experiment environment and apparatus: (a) monitor representing a user-facing flat surface; (b) printer and the surrounding wall representing far oblique surfaces; (c) bookshelf representing irregular surfaces; (d) lamp-shaped sculpture representing curved surfaces; (e) hat (for head tracking needed for Perspective Cursor); (f) mouse.

pose returns to geometry-based movement, and when applicable, the gap-filling tool can be used to bridge the gap.

5 EXPERIMENT

We conducted a controlled experiment to compare the performance of the geometry-based Everywhere Cursor technique with two perspective-based techniques: Perspective Cursor [18] and standard raycasting. Two tasks are used, target selection and tracing, on diverse geometric surfaces. We expect the geometry-based approach to have some advantages for detailed trajectory-based tasks due to its robustness against occlusion and minimal variation in CD gain on varying surfaces. For long-distance movements, we expect a perspective-based technique to perform better due to how a geometry-based approach requires full traversal of intervening geometry.

5.1 Participants

We recruited 12 participants who use the mouse with their right-hand (ages 20 to 33, heights between 163cm to 184cm, and 2 were female). All reported frequent mouse use (all more than 5 hours per day). Participants were recruited through online posting and word-of-mouth and received a $20 gift card.

5.2 SAR Environment

The SAR experiment environment is an office workspace including a large desk, other furniture, walls, and various objects (Figure 10). Four *main objects* serve as variations of surface geometry for short-distance tasks, with other objects creating representative variations in surface geometry to traverse over greater distances, including occlusion. The main objects are a monitor (1920 × 1080 24-inch display), a printer (40cm × 40cm × 25cm rectangular cuboid), a bookshelf with 6 books (heights 30 to 35cm, widths 2 to 5cm, and lengths 20 to 23cm), and a lamp-shaped sculpture (20cm-diameter sphere with 16cm-diameter cylinder base). All objects are approximately within arm's reach of the user (60cm), except for the printer, which is 2m away.

We have set up the physical environment to highlight common daily interactions in a SAR desktop environment with a wide range of surface geometries while providing a sufficient degree of control for internal validity (*i.e.*, source of differences in interaction behaviour can be analyzed according to key surface properties).

5.3 Apparatus

5.3.1 SAR Equipment. Three 1920 × 1200 60-FPS projectors and three cameras, all connected to a PC with a Quadro RTX 5000 display digital content. The layout minimizes occlusion while maintaining a 10% overlap between the projections (as recommended in RoomAlive [11]). The system generates SAR content using Unity 2019.4.4, where the position, rotation, and view frustum of each virtual camera in the game scene corresponds to a projector pose.

Christie Mystique software calibrates the projectors based on a static 3D scan of the room with 2mm resolution. The scan mesh is hole-filled and decimated to 10% of its original polygon count using VXElements with some manual cleanup using Blender. This process fills small holes up to approximately 1cm in diameter. Large gaps are "bridged" using the gap-filling technique described above. We intend to incorporate real-time room scanning in the future using depth cameras such as Microsoft Kinect [15].

5.3.2 6-DOF Mouse and Head Tracking. A thirteen-camera Vicon motion tracking system tracks the user's head and the mouse using 12mm markers. The participants wear a hat with five markers throughout the experiment for head tracking (Figure 10e). We attach a 10cm tall wooden structure with four markers to the mouse (Figure 10f showing a later 3D-printed version). Both structures weighed the same, with a total weight of mouse and structure 190g. The only difference is the fabrication method.

5.4 Techniques

Everywhere Cursor (geometry-based). The technique is implemented as described in the previous section with one important distinction. We instruct participants to use raycast clutching only for its intended purpose: to reset the cursor if it becomes stuck or lost. This change narrows the scope of the investigation to geometry-based movement instead of a mix of techniques. In practice, few participants used the feature (3 participants used it 3 times, and the rest used it less or not at all). Additionally, the experimenter used

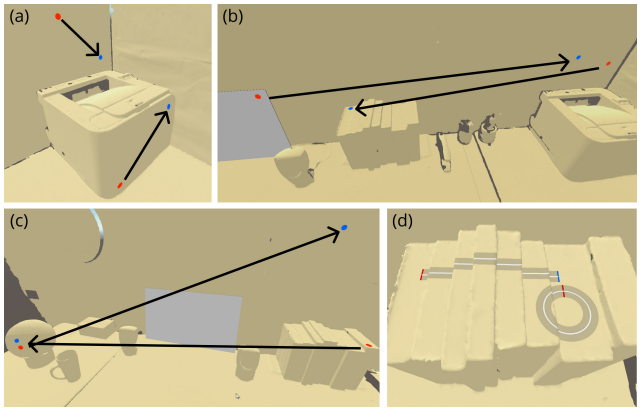

Figure 11: Example placement of experiment tasks: (a) short-distance selection; (b) long-distance selection with simple geometry; (c) long-distance selection with complex geometry; (d) linear and circular tracing.

gap-filling during the initial experiment setup, then the feature was disabled for participants.

Perspective Cursor (refined perspective-based). Our implementation of Perspective Cursor is a SAR-adaptation of the MDE technique developed by Nacenta et al. [18]. The system casts a virtual ray from the user's head to the cursor and maps mouse movement along the X- and Y-axes to the horizontal and vertical rotation of the ray. The intersection point of the ray with the scene 3D model determines the cursor position. The cursor visualization uses the same cursor projector as Everywhere Cursor. When the raycast does not intersect any surface, the cursor projector becomes hidden. However, the ray and mouse mapping behaviour are maintained, so movement out of the non-visible region is possible. Unlike the original implementation, we do not show a halo because most of our SAR environment is scanned and covered by the projection mapping, unlike in an MDE. This means there is little chance that the Perspective Cursor ray would not intersect a displayable surface. In pilot tests, we confirmed this to be the case, and when the rare event occurred, it did not hinder the user's interaction.

Raycasting (basic perspective-based). We use the same configuration and settings as Everywhere Cursor's raycast clutching, except the mouse need not be tilted, and the intersection point of the ray exclusively determines the cursor position.

We set the mouse pointer acceleration for Everywhere Cursor and Perspective Cursor to have the same speed on the monitor when the user is sitting normally on the chair. We log the cursor position at every frame for post hoc analysis for all techniques.

5.5 Tasks

Participants perform two tasks: target selection and tracing.

5.5.1 Selection. We display two 2cm-radius circular targets. The participants first click the red start target (upon which it turns gray to indicate a clicked state), and then click the blue end target next. The target colour becomes darker when the cursor is inside. We instruct the participants to select the targets as fast as possible without sacrificing accuracy. We record any clicks outside of the end target as errors, and the participant moves to the next task regardless.

For short-distance selection, we chose positions of target pairs to emphasize characteristic surface features while maintaining a 30cm 3D Euclidean distance between the two targets (Figure 11a). We use the Euclidean distance instead of distance along the surface because determining the latter on a non-planar and irregular geometry is challenging and hard to generalize. Long-distance selection is similar, but the targets are now 1.5m apart, and there are simple and complex geometry variations of the task. Simple tasks have mostly flat and obstruction-free surfaces between the targets (Figure 11b), while complex tasks have many intermediate objects requiring the cursor to traverse irregular geometry (Figure 11c). To remove visual search time when targets are spaced far apart, our system requires the participants to turn their heads to look at both targets before they are enabled for selection.

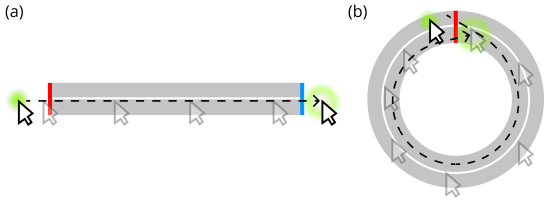

Figure 12: Illustration of tracing task: (a) linear (b) circular.

5.5.2 Tracing. We adapt a steering task [1] to evaluate linear and circular tracing. This task is a better measure of direct manipulation accuracy and precision when analyzing resulting trajectories [4]. In contrast, the target selection task focuses on speed and binary error rate as metrics, making a nuanced evaluation of accuracy and precision difficult.

For both linear and circular tracing, we show a 20mm × 200mm grey tunnel, along with a 4mm-wide white outline through its centre (Figure 12). The participants click down and drag the cursor through the tunnel, and we instruct them to prioritize accuracy over speed. If they release the click before moving past the end gate, our system requires them to repeat the task. While leaving the tunnel does not abort the trial, the tunnel turns red and an error sound plays to indicate that they must focus on tracing as accurately as possible.

We place the tunnels to reflect geometric features of each surface type (Figure 11d). Our tracing task follows the geometry (*i.e.* world-linear) since we focus on suface-mapped SAR where pixels on 2D textures are mapped to real surfaces, rather than making surfaces 'disappear'. We believe this is the most likely kind of SAR setup for desktop computing, where the system leverages geometric features of the real-world environment to configure the layout of virtual contents.

5.6 Procedure

The experiment consists of two stages: Stage 1 tests short-distance selections, and Stage 2 tests a mix of long-distance selections and tracing. In both stages, our system requires participants to sit in front of the monitor. We chose this order to familiarize participants with the techniques before doing more challenging tasks.

During Stage 1, participants perform short-distance selection tasks using the three techniques, one at a time. For each technique, the experimenter briefly explains its mechanism, gives a 2-minute instruction to explore the technique, and shows four target pairs for practice, one at a time. Afterward, the participants perform selection tasks in three blocks of four target pairs for each surface type, where the target pair order for each block is randomized. The system allows participants to rest at the end of each surface type.

In Stage 2, the experimenter explains how to perform long-distance selections and tracing. Then for each technique, the system displays one linear tunnel and one circular tunnel as practice, followed by a series of linear tracing, circular tracing, and long-distance selections during which we measure the participants' performance. The participants repeat this task series in three blocks. The system allows participants to rest at the end of each block.

5.7 Design

Each stage has a within-subjects design. Both have TECHNIQUE as its primary independent variable with 3 levels (EVERYWHERE, PERSPECTIVE, RAYCASTING). For short-distance selection and tracing, the secondary independent variable is SURFACE with 4 levels (NORMAL on monitor (Figure 10a), OBLIQUE on printer (Figure 10b), IRREGULAR on bookshelf (Figure 10c), CURVED on lamp (Figure 10d)). For long-distance selection, the secondary independent variable is TRAVERSAL with 4 levels (SIMPLE-R and SIMPLE-L for simple geometry, COMPLEX-L and COMPLEX-R for complex geometry).

In Stage 1, there are 4 task variations for each SURFACE. A randomized set of these task variations are repeated in 3 blocks per combination of TECHNIQUE and SURFACE. The order for TECHNIQUE is counter-balanced using a balanced Latin square, and for each technique, the order for SURFACE is randomized.

In Stage 2, there are 12 task variations for each TECHNIQUE: 1 linear and 1 circular tracing for each SURFACE, and 4 long-distance selections. The order for TECHNIQUE is counter-balanced using a Latin square. For each technique, the participants complete the task variations in the order of NORMAL tracings, SIMPLE-R selection, OBLIQUE tracings, SIMPLE-L selection, IRREGULAR tracings, COMPLEX-L selection, CURVED tracings, then COMPLEX-R selection. We use a fixed task order so the tasks increase in difficulty. This ordering was based on initial tests and pilot results which showed participants could more easily learn a technique when moving in smaller regions and simpler geometry. This series of tasks is repeated in 3 blocks per TECHNIQUE.

The primary measures computed from logs are *Selection Time* and *Selection Error* for selection tasks, and *Tracing Time*, *Tracing Error* and *Path Deviation* for tracing tasks. *Selection Time* is defined as the duration starting at the moment the start target is clicked until a click-down event. *Selection Error* is the proportion of trials with errors, reported as an error rate. *Tracing Time* is the duration from crossing the start gate to the end gate. *Tracing Error* is a measure of accuracy calculated as the average distance orthogonal to the tracing path (for linear) or radial distance (for circular) away from the ideal path, as per its definition by Cami et al. [4]. *Path Deviation* is the standard deviation of the cursor distance orthogonal to the tracing path (for linear) or its radial distance from the origin (for circular).

In addition, a post-technique questionnaire provides 3 subjective measures for measuring satisfaction, intuitiveness, and physical fatigue. All ratings were worded similarly (*e.g.* "how satisfied were you with this technique for this task," where "satisfied" was changed for the other 2 measures) and used a 7-point numerical scale, with the extremes labelled as "not at all" for 1 and "very" for 7. At the end of the experiment, participants gave written comments on their experience with each technique.

In summary: 3 TECHNIQUES × 4 SURFACES × 3 BLOCKS × 4 target pairs = 144 data points per participant for Stage 1, and 3 TECHNIQUES × 4 SURFACES × 3 BLOCKS × 2 tracings = 72 data points per participants for Stage 2.

6 RESULTS

In the analysis to follow, a TECHNIQUE × SURFACE ANOVA with Tukey HSD post hoc tests was used, unless noted otherwise. When the assumption of sphericity was violated, degrees of freedom were corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$). According to the Shapiro-Wilks Normality test, none of the residuals for the measured data were normally distributed, so Box-Cox or ART-transformed [31] values were used for statistical analysis. For each measure, trials were aggregated by participant and factors being analyzed. *To streamline the presentation of results, all statistical test details are provided as tables in Appendix A. References are in the form "A.1: Table 1a" where A.1 refers to subsection 1 of the Appendix.* Data and analysis scripts are available[2].

Outliers. For each combination of participant, TECHNIQUE, and TRAVERSAL (long-distance selection) or SURFACE (all others), task times more than 3 standard deviations from the mean time were excluded as outliers. In total, 24 trials (1.4%) were removed for short-distance selection, 2 trials (0.4%) for long-distance selection, and none for linear and circular tracing.

Learning Effects. We are interested in practised performance, so we examine if earlier blocks took longer and should be removed. For the selection tasks, there is a significant main effect for BLOCK on *Selection Time* for short-distance selection ($F_{2,22} = 37.42$, $p < .0001$, $\eta^2_G = .10$) and long-distance selection ($F_{2,22} = 6.60$, $p < .001$, $\eta^2_G = .09$), but no interaction effects involving BLOCK. Post hoc tests found block 1 significantly slower than block 3 (all $p < .05$). In subsequent analysis, blocks 2 and 3 are used for selection tasks since they are more representative of practised performance [29]. For the steering tasks, no main effect was found for BLOCK on *Tracing Time*, and no interaction effects involving BLOCK. Therefore, all blocks are used in the following analysis.

6.1 Short-distance Selection

Selection Time. Everywhere Cursor and Perspective Cursor were both faster than Raycasting (Figure 13a, A.1: Table 2a). We observed up to a 33% time savings for the mouse-based techniques compared to RAYCASTING. Perspective Cursor was faster than Everywhere Cursor on irregular surfaces, while they were comparable on other surfaces (A.1: Table 2b). We observed up to a 30% reduction in time for the mouse-based techniques compared to RAYCASTING on NORMAL, OBLIQUE, and IRREGULAR surfaces, and a 40% reduction

[2]https://github.com/exii-uw/sar-mouse-cursor-control

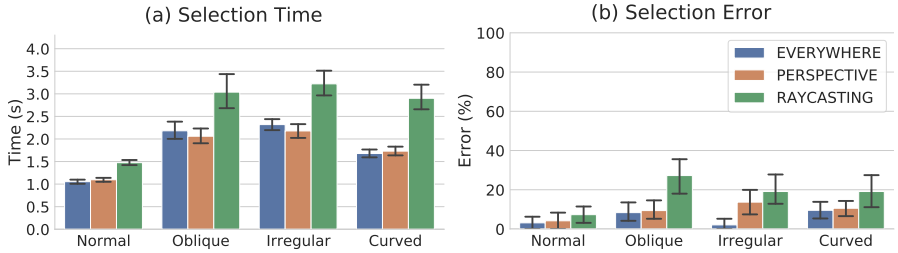

Figure 13: Short-distance selection task (a) *Selection Time* (b) *Selection Error* by TECHNIQUE for each SURFACE (error bars in all graphs are 95% confidence).

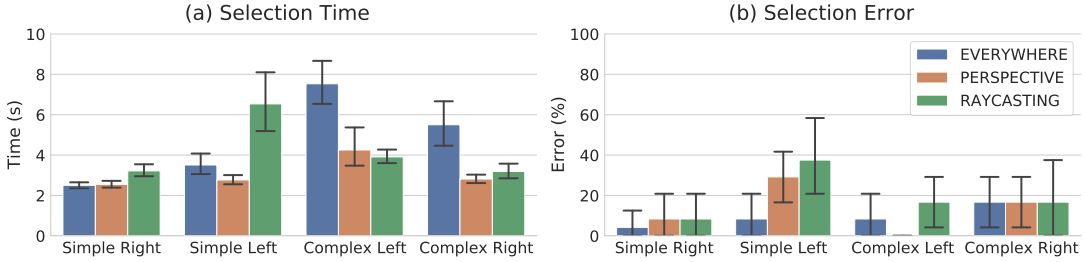

Figure 14: Long-distance selection task (a) *Selection Time* (b) *Selection Error* by TECHNIQUE for each TRAVERSAL.

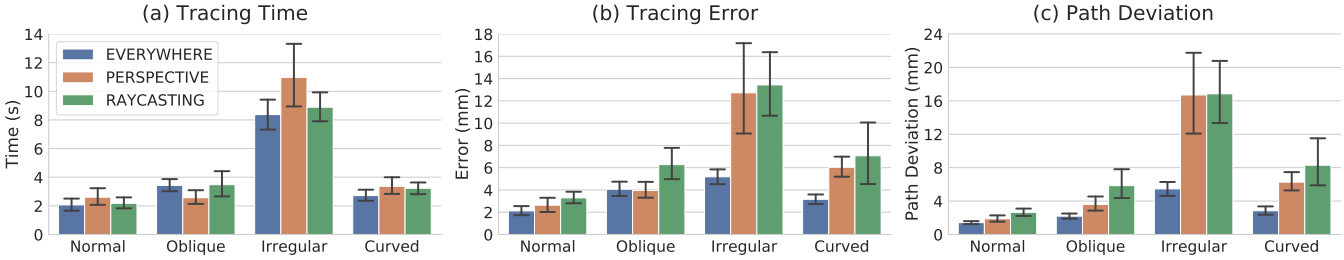

Figure 15: Linear tracing task (a) *Tracing Time* (b) *Tracing Error* (c) *Path Deviation* by TECHNIQUE for each SURFACE.

on CURVED surfaces. Additionally, there was a 6% reduction for Perspective Cursor compared to Everywhere Cursor on IRREGULAR surfaces.

Selection Error. Everywhere Cursor and Perspective Cursor are less error-prone than Raycasting (Figure 13b, A.1: Table 3a). We observed up to a 12% reduction in error rate for EVERYWHERE and a 9% reduction for PERSPECTIVE compared to RAYCASTING. Everywhere Cursor is less error-prone than perspective-based techniques on irregular surfaces, and all techniques have comparable error rates on normal and curved surfaces (Figure 13b, A.1: Table 3b). We observed up to a 17% reduction for EVERYWHERE compared to perspective-based techniques on IRREGULAR surfaces.

6.2 Long-distance Selection

Selection Time. Over long distances, Perspective Cursor was faster than Everywhere Cursor and Raycasting (Figure 14a, A.2: Table 4a). We observed up to a 35% time savings for PERSPECTIVE compared to EVERYWHERE, and a 27% time savings compared to RAYCASTING. Everywhere Cursor and Perspective Cursor were faster than Raycasting when traversing simple geometries, but Perspective Cursor and Raycasting are faster than Everywhere Cursor when traversing complex geometries (A.2: Table 4b). We observed up to a 22% reduction in time for the mouse-based techniques compared to RAYCASTING when traversing simple geometry. Moreover, there was up to a 49% reduction for PERSPECTIVE and RAYCASTING compared to EVERYWHERE when traversing complex geometry.

Selection Error. Everywhere Cursor was more accurate than Perspective Cursor and Raycasting (Figure 14b, A.2: Table 5a). We observed up to a 4% reduction in error rate for EVERYWHERE compared to PERSPECTIVE, and a 10% reduction compared to RAYCASTING. While there was an interaction between TECHNIQUE and TRAVERSAL on *Selection Error* (A.2: Table 5b), the post hoc tests were not able to show meaningful differences between conditions, likely due to insufficient statistical power.

6.3 Linear Tracing

Tracing Time. Overall, all techniques had comparable speeds, but Perspective Cursor was faster than Everywhere Cursor on oblique surfaces (Figure 15a, A.3: Table 6). We observed up to a 25% reduction in time for PERSPECTIVE compared to other techniques.

Tracing Error. Everywhere Cursor was more accurate than Perspective Cursor and Raycasting, and Perspective Cursor was more accurate than Raycasting (Figure 15b, A.3: Table 7a). We observed up to

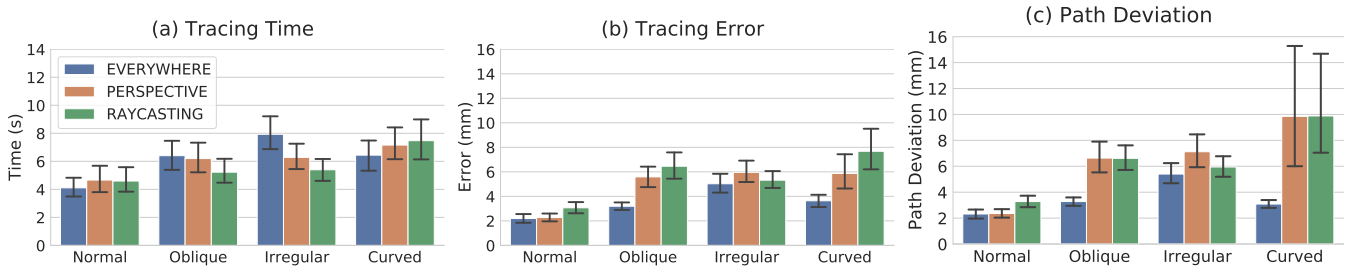

Figure 16: Circular tracing task (a) *Tracing Time* (b) *Tracing Error* (c) *Path Deviation* by TECHNIQUE for each SURFACE.

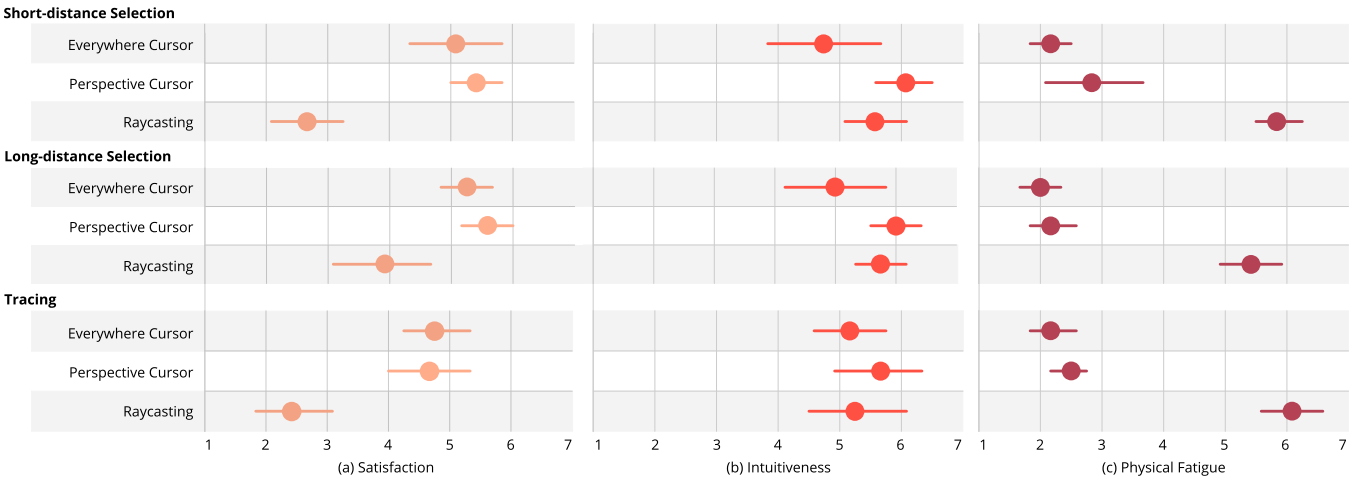

Figure 17: (a) *Satisfaction* (b) *Intuitiveness* (c) *Physical Fatigue* by TECHNIQUE for each task.

a 43% reduction in error for EVERYWHERE compared to PERSPECTIVE, and a 52% reduction compared to RAYCASTING. Additionally, there was a 16% reduction for PERSPECTIVE compared to RAYCASTING. Everywhere Cursor and Perspective Cursor have comparable accuracy on normal and oblique surfaces, but Everywhere Cursor is more accurate on irregular and curved surfaces (A.3: Table 7b). We observed up to a 60% reduction in error for EVERYWHERE compared to PERSPECTIVE on IRREGULAR and CURVED surfaces.

Path Deviation. Everywhere Cursor was more precise than Perspective Cursor and Raycasting, and Perspective Cursor more precise than Raycasting (Figure 15c, A.3: Table 8a). We observed up to a 58% reduction in path deviation for EVERYWHERE compared to PERSPECTIVE, and a 65% reduction compared to RAYCASTING. Additionally, there was a 15% reduction for PERSPECTIVE compared to RAYCASTING.

6.4 Circular Tracing

Tracing Time. Raycasting was faster than Perspective Cursor on irregular surfaces, with up to a 32% reduction in time for RAYCASTING compared to EVERYWHERE on IRREGULAR surfaces (Figure 16a, A.4: Table 9b).

Tracing Error. Everywhere Cursor was more accurate than Perspective Cursor and Raycasting, and Perspective Cursor more accurate than Raycasting (Figure 16b, A.4: Table 10a). We observed up to a 29% reduction in error for EVERYWHERE compared to PERSPECTIVE, and a 38% reduction compared to RAYCASTING. Moreover, there was a 13% reduction for PERSPECTIVE compared to RAYCASTING. Everywhere Cursor and Perspective Cursor had comparable accuracy on normal and irregular surfaces, but Everywhere Cursor was more accurate than Perspective Cursor on oblique and curved surfaces (A.4: Table 10b). We observed up to a 55% reduction for EVERYWHERE compared to PERSPECTIVE on OBLIQUE and CURVED surfaces.

Path Deviation. Everywhere Cursor was more precise than Perspective Cursor and Raycasting (Figure 16c, A.4: Table 11a). We observed up to a 46% reduction in path deviation for EVERYWHERE compared to other techniques. Everywhere Cursor and Perspective Cursor had comparable precision on normal and irregular surfaces, but Everywhere Cursor was more precise than Perspective Cursor on oblique and curved surfaces (A.4: Table 11b). We observed up to a 50% reduction for EVERYWHERE compared to PERSPECTIVE on SLANTED surfaces, and a 70% reduction on CURVED surfaces.

6.5 Subjective Ratings

Short-distance Selection. Everywhere Cursor and Perspective Cursor were comparable across all subjective ratings, while Raycasting was less satisfactory and caused higher fatigue (Figure 17a, A.5: Table 12).

Long-distance Selection. The results are similar to short-distance selections, except that Everywhere Cursor was rated as less intuitive (Figure 17b, A.5: Table 13).

Tracing. Again, Everywhere Cursor and Perspective Cursor were comparable across all measures, and Raycasting was rated worse in all measures but was comparably intuitive to use (Figure 17c, A.5: Table 14).

7 DISCUSSION

We first provide a high-level comparison of techniques based on the experiment results. Then we discuss potential applications for mouse interactions in SAR, limitations of our study, and areas for future work.

The geometry-based Everywhere Cursor produces more precise and accurate trajectories than perspective-based techniques when the surface geometry deviates from a conventional flat user-facing surface. For instance, one of the participants wrote that "[t]racing over surfaces that are not flat [is] almost impossible" with Perspective Cursor and Raycasting. The reason is likely because a geometry-based technique maintains a consistent CD gain. With a perspective-based technique, the gain changes unpredictably depending on the distance to the surface. For surfaces far away or with deep depth deviation, this variation in CD gain significantly hinders a precise and consistent control over the cursor's speed and movement. One participant indicated that Everywhere Cursor is "great for going over edges" and "precise control over single surfaces," which further supports this explanation and shows users value accurate movement in a complex environment.

Environmental occlusion also has a pronounced effect on the performance of perspective-based techniques, while Everywhere Cursor's movement remained robust. Irregular surfaces, for example, had many occluding peaks and corners, which produced erratic cursor trajectory with perspective-based techniques. Other environmental objects, such as the puzzle box in the oblique surface condition, frequently masked the interactive surfaces. Similarly, cursor movements near the edges of objects often made the cursor jump between the interacting surface and the background. In contrast, occlusion does not affect Everywhere Cursor, which allows the cursor to maintain a smooth and sensible trajectory even in the presence of intermittent occlusion.

Participants also appreciate how geometry-based behaviour resembled a traditional desktop cursor across different surface types: questionnaire responses show that using Everywhere Cursor is "easy and simple," particularly on curved and edged surfaces, and that it behaves "similar to [their] daily experience when using a mouse on different monitors."

Our results show a pronounced effect of the difference in precision and accuracy in the tracing tasks, but selection tasks seem unaffected by this behaviour with the mouse-based techniques. This is likely because the accuracy of the cursor path during a ballistic movement has less bearing on pointing performance. The accuracy is primarily determined during the final corrective period near the end target, where the movement range is likely too small to demonstrate the impact of varying gain or occlusion.

Likely due to its geometry-following nature, Everywhere Cursor is slower than perspective-based techniques when the ballistic movement traverses complex geometry. Everywhere Cursor assumes a spatial understanding of the scene geometry to make long-distance ballistic movements efficient. Essentially, the user must maneuver around obstructions and large gaps, which lead to longer movements and a higher cognitive load to plan geometrically optimal movements. One participant felt that Everywhere Cursor is "least effective for long distance" due to the need for finding complicated "connected paths," which further explains the slow selection time for long-distance targets.

To summarize, the geometry-based Everywhere Cursor performs well with detailed interactions focused on one local area at a time. In contrast, perspective-based methods like Perspective Cursor and Raycasting shine in situations where large-scale movement or manipulation is important and absolute precision or accuracy is not a significant concern.

7.1 Applications

We describe a scenario that illustrates some examples of mouse interaction in SAR (Figure 18), and we provide an analysis regarding where perspective-based and geometry-based approaches would be most beneficial.

Scenario. Alice, the lead designer of an ocean-themed immersive art event, walks into the venue. At the centre, there is a desk with the venue's main architectural piece in front of it, where she intends to design the projection mapping content on. She places her laptop on the desk, automatically expanding her desktop environment and active applications throughout the room. She brings the mouse cursor from the monitor to the lamp on her desk, where she scrolls through the cover flow of her favourite albums. After putting on her work playlist, she moves the focus to the bookshelf. The books display her quick-access applications. She moves the cursor slightly around the book spine showing a 3D design app and takes a peek at the available project files. She opens the appropriate project, initializing the room with projection-mapped art content.

She wants to put a swarm of fish swimming over the stairs as the beginning hook of the experience, so she moves the cursor from the laptop over to the stairs and drags to create an animation path. Next, she notices that the curved part of the wall has images that do not fit with the rest of the aesthetics, so she lasso-selects and deletes them. She then drags the cursor from one end of the wall to the other to define an effect that fills the wall with a blue hue, giving the audience a feeling of being in the ocean. After this, she is satisfied with the results.

Just then, her phone rings to let her know that she has a meeting coming up soon. She quits the program and prints her presentation material by dragging-and-dropping it from her laptop to the printer.

Analysis. Among the illustrated mouse interactions, the trajectory-based tasks like defining animation paths (Figure 18d, f) and lasso-selection, especially on curved surfaces (Figure 18e), are most appropriate for Everywhere Cursor given its robust ability to produce continuous paths. Furthermore, other interactions on irregular (Figure 18b) or curved surfaces (Figure 18a) are also likely to benefit from Everywhere Cursor's strength against intermittent occlusion. In contrast, mouse movement across separated surfaces, such as moving from the monitor to the wall (Figure 18c) or drag-and-drop to the printer (Figure 18g), are most suited for Perspective Cursor, since it can move the cursor quickly across long distances,

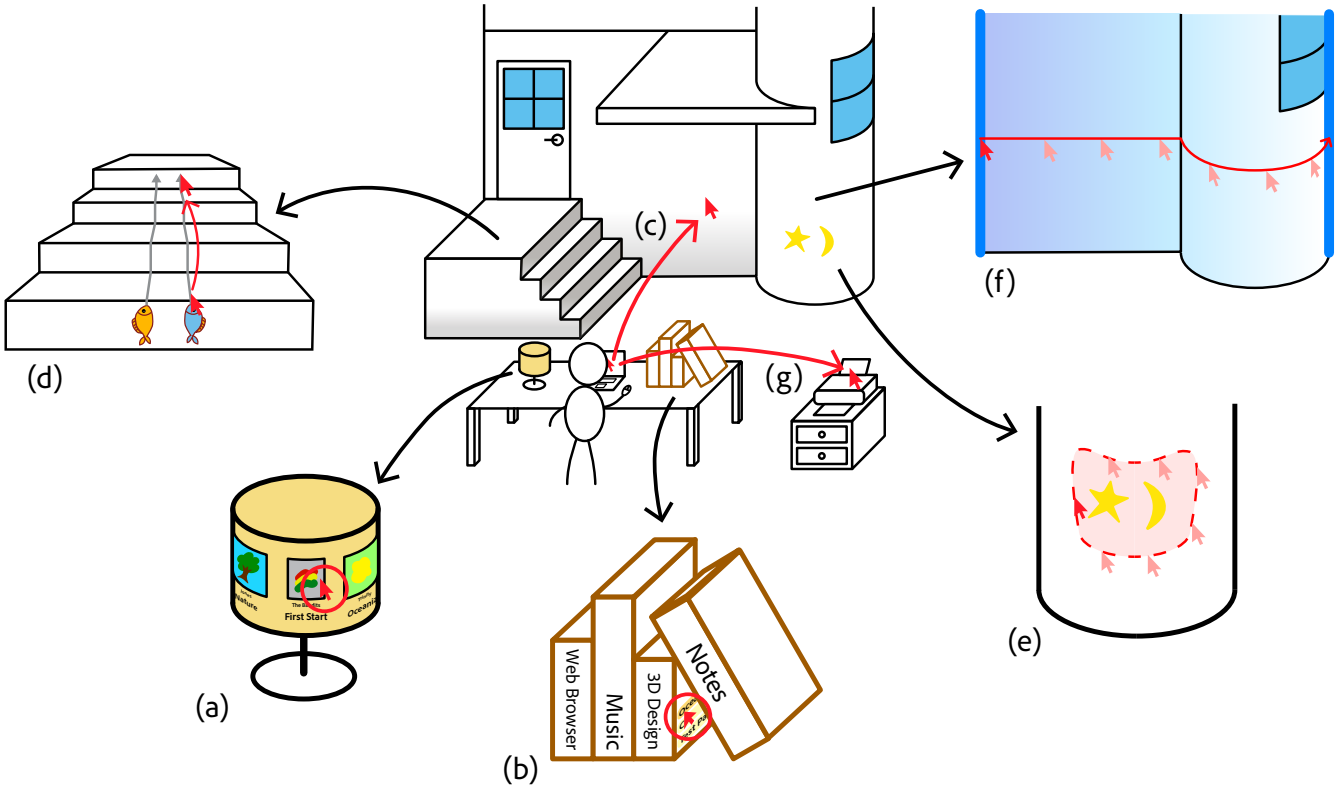

Figure 18: Example scenario of a SAR desktop: (a) using a desk lamp as a surface for music album browsing; (b) repurposing a bookshelf as a quick-access menu; (c) changing interaction surface from laptop screen to the building wall; (d) defining animation path for virtual objects; (e) lasso-selection on a curved wall; (f) defining a tweened animation effect over the wall; (g) drag-and-drop onto the printer.

and accuracy or precision holds less importance in these types of interaction.

7.2 Limitations

We did not experimentally control the cursor path in long-distance selections. The participants were free to move the cursor however they wanted to as long as they completed the tasks. We made this choice to ensure a reasonable degree of external validity from realistic interaction scenarios. Participants chose many different paths when using Everywhere Cursor, encountering very different sets of obstacles. Therefore, it is difficult to identify factors affecting their interaction behaviour rigorously. Examining what types of obstructive geometry causes a significant drop in performance with greater control may provide additional insight into the techniques.

The mouse weight may have reduced Raycasting performance. Some participants commented that the mouse felt heavy. However, the mouse weighs less than a typical VR controller (*e.g.* 190g compared to ~200g for the HTC VIVE controller). A reasonable explanation could be that a mouse is not ergonomically designed to hold in mid-air to raycast, causing fatigue [18, 30]. Moreover, our results for short-distance selection on the monitor show that Raycasting can be accurate and fast when conditions are fully optimal, which is consistent with Nacenta et al.'s optimal *within-display* condition [18].

Despite predominantly large effect sizes for main effects [2] ($\eta_G{}^2$ between 0.27 and 0.84), our data may not capture smaller differences without more participants. Effect sizes for interactions were small to medium [2] ($\eta_G{}^2$ between 0.05 and 0.17, except long-distance selection time with 0.34). Notably, despite a significant interaction with long-distance error rates, subsequent post hoc tests revealed no pairwise differences.

Lastly, our study has relatively low diversity in gender and age, which could limit the generalizability of our findings, such as to older adults or those with poor motor skills. Further exploration with a more diverse population could yield more detailed insights.

7.3 Future Work

While Everywhere Cursor included direct raycasting as a clutching mechanism, our results indicate that raycasting is inappropriate for frequent use due to the high rate of error and fatigue. In contrast, Perspective Cursor offers strong performance for high-speed, long-distance movements, and it is relatively comfortable to use since the user does not need to lift the mouse. Therefore, a speed-based mixing of perspective and geometry approaches could make interactions more optimal, specifically by using Everywhere Cursor for fine details at low speed and Perspective Cursor at high speeds across large surfaces. This could be done by linearly interpolating the resulting positions from the two techniques and spherically

interpolating the resulting cursor orientations. The interpolation parameter can be obtained from $t(\vec{v}) = \frac{1}{1+e^{-k(\|\vec{v}\|-c)}} \in (0,1)$, which is a logistic function in terms of mouse delta vector $\vec{v} \in \mathbb{R}^2$, where parameters k and c define growth rate and horizontal center of the function curve, respectively. With this interpolation, we can avoid abrupt changes in the cursor movement while achieving the mixing of the two behaviours as intended.

Recall that Everywhere Cursor only adjusts the cursor's upright orientation after movement is idle for 0.5s. No participant commented on this behaviour, but future work could examine more open-ended tasks over irregular surfaces to see if that exposes issues with this behaviour.

Future work can also explore methods to automate gap-filling. Currently, the Everywhere Cursor method relies on a user-initiated interactive method to bridge large gaps. It may be possible to extend existing hole-filling algorithms to identify and "bridge" these types of gaps automatically. For example, a machine learning approach could leverage training data from the current manual technique. The challenge is to identify pairs of surfaces the user will most likely traverse and how to place the bridging geometry.

Finally, while we focused on a realistic office environment, testing cursor techniques in multiple, diverse environments could reveal interesting insights into the generalizability of our results across varying types of geometry. For instance, we may encounter a different or more nuanced set of design challenges in a larger-scale projection-mapping venue or a space with a different set of objects.

8 CONCLUSION

We explored two approaches to mouse cursor control in SAR, perspective-based and geometry-based. Techniques using perspective-based approaches have been proposed for MDEs and limited forms of SAR, but the Everywhere Cursor technique we introduced in this paper is the first to uses a geometry-based approach focused to address design challenges for SAR cursor control. In a controlled experiment examining selection and tracing across short and long distances, we compared the performance of this geometry-based technique with two representative perspective-based techniques, Perspective Cursor and Raycasting. Our results show the geometry-based Everywhere Cursor is highly effective in producing a precise and accurate trajectory, but the perspective-based techniques are faster for distant target selection across large regions of complex geometry. Based on our examination, we discuss beneficial use cases for a desktop-like mouse interaction in SAR, such as intuitive projection-mapping content design and converting everyday objects into a platform for digital interactive media. We hope our work inspires others to examine how familiar techniques, like a mouse-controlled cursor, can be adapted to enable intuitive interaction with emerging computing environments.

ACKNOWLEDGMENTS

This work made possible by NSERC Discovery Grant 2018-05187, Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces," Ontario Early Researcher Award ER16-12-184, NSERC USRA Program for undergraduate research, and the Cheriton School of Computer Science Undergraduate Research Fellowship.

REFERENCES

[1] Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '97)*. Association for Computing Machinery, New York, NY, USA, 295–302. https://doi.org/10.1145/258549.258760

[2] Roger Bakeman. 2005. Recommended effect size statistics for repeated measures designs. *Behavior research methods* 37, 3 (2005), 379–384.

[3] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. *RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300331

[4] Drini Cami, Fabrice Matulic, Richard G. Calland, Brian Vogel, and Daniel Vogel. 2018. Unimanual Pen+Touch Input Using Variations of Precision Grip Postures. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 825–837. https://doi.org/10.1145/3242587.3242652

[5] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 € Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639

[6] Renaud Gervais, Jérémy Frey, and Martin Hachet. 2015. Pointing in Spatial Augmented Reality from 2D Pointing Devices. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Springer International Publishing, Cham, 381–389.

[7] Renaud Gervais, Joan Sol Roo, and Martin Hachet. 2016. Tangible Viewports: Getting Out of Flatland in Desktop Environments. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (Eindhoven, Netherlands) *(TEI '16)*. Association for Computing Machinery, New York, NY, USA, 176–184. https://doi.org/10.1145/2839462.2839468

[8] Jeremy Hartmann and Daniel Vogel. 2021. An examination of mobile phone pointing in surface mapped spatial augmented reality. *International Journal of Human-Computer Studies* 153 (2021), 102662. https://doi.org/10.1016/j.ijhcs.2021.102662

[9] Hiroshi Ishii, Craig Wisneski, Scott Brave, Andrew Dahley, Matt Gorbet, Brygg Ullmer, and Paul Yarin. 1998. AmbientROOM: Integrating Ambient Media with Architectural Space. In *CHI 98 Conference Summary on Human Factors in Computing Systems* (Los Angeles, California, USA) *(CHI '98)*. Association for Computing Machinery, New York, NY, USA, 173–174. https://doi.org/10.1145/286498.286652

[10] Brad Johanson, Greg Hutchins, Terry Winograd, and Maureen Stone. 2002. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France) *(UIST '02)*. Association for Computing Machinery, New York, NY, USA, 227–234. https://doi.org/10.1145/571985.572019

[11] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 637–644. https://doi.org/10.1145/2642918.2647383

[12] Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. 2010. A Comparison of Ray Pointing Techniques for Very Large Displays. In *Proceedings of Graphics Interface 2010* (Ottawa, Ontario, Canada) *(GI '10)*. Canadian Information Processing Society, CAN, 269–276.

[13] Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, Beverly Harrison, Adam D. Rea, Matthai Philipose, and Anthony LaMarca. 2009. Bonfire: A Nomadic System for Hybrid Laptop-Tabletop Interaction. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (Victoria, BC, Canada) *(UIST '09)*. Association for Computing Machinery, New York, NY, USA, 129–138. https://doi.org/10.1145/1622176.1622202

[14] Daekun Kim and Daniel Vogel. 2022. Everywhere Cursor: Extending Desktop Mouse Interaction into Spatial Augmented Reality. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts)* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, 9 pages. https://doi.org/10.1145/3491101.3519796

[15] David Lindlbauer and Andy D. Wilson. 2018. Remixed Reality: Manipulating Space and Time in Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173703

[16] Blair MacIntyre, Elizabeth D. Mynatt, Stephen Voida, Klaus M. Hansen, Joe Tullio, and Gregory M. Corso. 2001. Support for Multitasking and Background Awareness Using Interactive Peripheral Displays. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) *(UIST '01)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/502348.502355

[17] Michael R. Marner, Ross T. Smith, and Bruce H. Thomas. 2015. Mapping 2D input to 3D immersive spatial augmented reality. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. Institute of Electrical and Electronics Engineers, Arles, France, 171–172. https://doi.org/10.1109/3DUI.2015.7131755

[18] Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin. 2006. Perspective Cursor: Perspective-Based Interaction for Multi-Display Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 289–298. https://doi.org/10.1145/1124772.1124817

[19] Kousuke Nakashima, Takashi Machida, Kiyoshi Kiyokawa, and Haruo Takemura. 2005. A 2D-3D Integrated Environment for Cooperative Work. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Monterey, CA, USA) *(VRST '05)*. Association for Computing Machinery, New York, NY, USA, 16–22. https://doi.org/10.1145/1101616.1101621

[20] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. 2016. Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing* (San Francisco, California, USA) *(CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 1716–1725. https://doi.org/10.1145/2818048.2819965

[21] Simon T. Perrault, Eric Lecolinet, Yoann Pascal Bourse, Shengdong Zhao, and Yves Guiard. 2015. Physical Loci: Leveraging Spatial, Object and Semantic Memory for Command Selection. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 299–308. https://doi.org/10.1145/2702123.2702126

[22] Claudio S. Pinhanez. 2001. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In *Proceedings of the 3rd International Conference on Ubiquitous Computing* (Atlanta, Georgia, USA) *(UbiComp '01)*. Springer-Verlag, Berlin, Heidelberg, 315–331.

[23] Christian Pirchheim, Manuela Waldner, and Dieter Schmalstieg. 2009. Deskotheque: Improved Spatial Awareness in Multi-Display Environments. In *2009 IEEE Virtual Reality Conference*. Institute of Electrical and Electronics Engineers, Lafayette, Louisiana, 123–126. https://doi.org/10.1109/VR.2009.4811010

[24] Ramesh Raskar, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. 2003. ILamps: Geometrically Aware and Self-Configuring Projectors. *ACM Trans. Graph.* 22, 3 (jul 2003), 809–818. https://doi.org/10.1145/882262.882349

[25] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 179–188. https://doi.org/10.1145/280814.280861

[26] Jun Rekimoto. 1997. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (Banff, Alberta, Canada) *(UIST '97)*. Association for Computing Machinery, New York, NY, USA, 31–39. https://doi.org/10.1145/263407.263505

[27] Jun Rekimoto and Masanori Saitoh. 1999. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 378–385. https://doi.org/10.1145/302979.303113

[28] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction* (Rome, Italy) *(INTERACT'05)*. Springer-Verlag, Berlin, Heidelberg, 267–280. https://doi.org/10.1007/11555261_24

[29] R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI. *Int. J. Hum.-Comput. Stud.* 61, 6 (dec 2004), 751–789. https://doi.org/10.1016/j.ijhcs.2004.09.001

[30] Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) *(UIST '05)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1095034.1095041

[31] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[32] Robert Xiao, Miguel A. Nacenta, Regan L. Mandryk, Andy Cockburn, and Carl Gutwin. 2011. Ubiquitous Cursor: A Comparison of Direct and Indirect Pointing Feedback in Multi-Display Environments. In *Proceedings of Graphics Interface 2011* (St. John's, Newfoundland, Canada) *(GI '11)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 135–142.

[33] Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 354, 17 pages. https://doi.org/10.1145/3491102.3501884

A TABLES OF STATISTICAL TESTS

This appendix presents tables of ANOVA and post hoc statistical tests for main effects and interactions for dependent measures discussed in Results (Section 6). The section names correspond to those used in Results.

A.1 Short-distance Selection

Table 2: *Selection Time* for short-distance selections

(a) TECH ($F_{2,22} = 97.31$, $p < .0001$, $\eta^2_G = 0.68$)				
comparisons		*diff (ms)*	*p-value*	
EVERYWHERE	PERSPECTIVE	41	.70	
EVERYWHERE	RAYCASTING	-850	< .0001	***
PERSPECTIVE	RAYCASTING	-891	< .0001	***
(b) TECH × SURFACE ($F_{6,66} = 5.62$, $p < .0001$, $\eta^2_G = 0.10$)				
comparisons for NORMAL		*diff (ms)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-43	.095	
EVERYWHERE	RAYCASTING	-421	< .0001	***
PERSPECTIVE	RAYCASTING	-379	< .0001	***
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	120	.37	
EVERYWHERE	RAYCASTING	-857	< .0001	***
PERSPECTIVE	RAYCASTING	-977	< .0001	***
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	140	.014	*
EVERYWHERE	RAYCASTING	-903	< .0001	***
PERSPECTIVE	RAYCASTING	-1043	< .0001	***
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-53	.66	
EVERYWHERE	RAYCASTING	-1224	< .0001	***
PERSPECTIVE	RAYCASTING	-1171	< .0001	***

Table 3: *Selection Error* for short-distance selections

(a) TECH ($F_{2,22} = 22.04$, $p < .0001$, $\eta^2_G = 0.27$)				
comparisons		*diff (%)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-3.7	.046	*
EVERYWHERE	RAYCASTING	-12.4	< .0001	***
PERSPECTIVE	RAYCASTING	-8.7	.003	**
(b) TECH × SURFACE ($F_{6,66} = 2.98$, $p < .05$, $\eta^2_G = 0.09$)				
comparisons for NORMAL		*diff (%)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-1.0	1.00	
EVERYWHERE	RAYCASTING	-4.2	1.00	
PERSPECTIVE	RAYCASTING	-3.1	1.00	
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	-1.0	1.00	
EVERYWHERE	RAYCASTING	-18.9	0.009	**
PERSPECTIVE	RAYCASTING	-17.9	0.022	*
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	-11.6	.026	*
EVERYWHERE	RAYCASTING	-17.0	< .0001	***
PERSPECTIVE	RAYCASTING	-5.4	1.00	
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-1.0	.66	
EVERYWHERE	RAYCASTING	-9.6	< .0001	***
PERSPECTIVE	RAYCASTING	-8.5	< .0001	***

A.2 Long-distance Selection

Table 4: *Selection Time* for long-distance selections

(a) TECH ($F_{2,22} = 21.95$, $p < .0001$, $\eta^2_G = 0.39$)				
comparisons		*diff (ms)*	*p-value*	
EVERYWHERE	PERSPECTIVE	1666	< .0001	***
EVERYWHERE	RAYCASTING	-850	0.44	
PERSPECTIVE	RAYCASTING	-1117	< .0001	***
(b) TECH × TRAVERSAL ($F_{6,66} = 15.68$, $p < .0001$, $\eta^2_G = 0.34$)				
comparisons for SIMPLE-R		*diff (ms)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-45	.78	
EVERYWHERE	RAYCASTING	-713	< .0001	***
PERSPECTIVE	RAYCASTING	-668	.0002	***
comparisons for SIMPLE-L				
EVERYWHERE	PERSPECTIVE	740	.11	
EVERYWHERE	RAYCASTING	-3027	< .0001	***
PERSPECTIVE	RAYCASTING	-3768	< .0001	***
comparisons for COMPLEX-L				
EVERYWHERE	PERSPECTIVE	3280	< .0001	***
EVERYWHERE	RAYCASTING	3622	< .0001	***
PERSPECTIVE	RAYCASTING	343	< .97	
comparisons for COMPLEX-R				
EVERYWHERE	PERSPECTIVE	2689	< .0001	***
EVERYWHERE	RAYCASTING	2316	.0007	***
PERSPECTIVE	RAYCASTING	-373	< .33	

Table 5: *Selection Error* for long-distance selections

(a) TECH ($F_{2,22} = 4.88$, $p < .001$, $\eta^2_G =$)

comparisons		*diff (%)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-4.2	.029	*
EVERYWHERE	RAYCASTING	-10.4	.014	*
PERSPECTIVE	RAYCASTING	-6.3	.68	

(b) TECH × TRAVERSAL ($F_{6,66} = 2.78$, $p < .05$, $\eta^2_G =$)

n/a

A.3 Linear Tracing

Table 6: *Tracing Time* for linear tracing

(a) TECH ($F_{2,22} = 0.50$, $p = .06$, $\eta^2_G = 0.01$)

comparisons		*diff (ms)*	*p-value*	

n/a

(b) TECH × SURFACE ($F_{6,66} = 5.62$, $p < .0001$, $\eta^2_G = 0.10$)

comparisons for NORMAL		*diff (ms)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-541	.45	
EVERYWHERE	RAYCASTING	-113	.87	
PERSPECTIVE	RAYCASTING	427	.87	
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	863	.028	*
EVERYWHERE	RAYCASTING	-49	.34	
PERSPECTIVE	RAYCASTING	-912	.34	
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	-2598	.15	
EVERYWHERE	RAYCASTING	-514	.48	
PERSPECTIVE	RAYCASTING	2083	.41	
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-647	.25	
EVERYWHERE	RAYCASTING	-512	.25	
PERSPECTIVE	RAYCASTING	134	.99	

Table 7: *Tracing Error* for linear tracing

(a) TECH ($F_{2,22} = 20.41$, $p < .0001$, $\eta^2_G = 0.34$)

comparisons		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-2.7	.0004	***
EVERYWHERE	RAYCASTING	-3.9	< .0001	***
PERSPECTIVE	RAYCASTING	-1.2	.007	**

(b) TECH × SURFACE ($F_{6,66} = 4.39$, $p < .001$, $\eta^2_G = 0.09$)

comparisons for NORMAL		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.5	.40	
EVERYWHERE	RAYCASTING	-1.2	.0012	**
PERSPECTIVE	RAYCASTING	-0.7	.011	*
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	-0.1	.58	
EVERYWHERE	RAYCASTING	-2.2	.013	*
PERSPECTIVE	RAYCASTING	-2.3	.0036	**
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	-7.5	< .0001	***
EVERYWHERE	RAYCASTING	-8.2	< .0001	***
PERSPECTIVE	RAYCASTING	-0.7	.17	
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-2.9	< .0001	***
EVERYWHERE	RAYCASTING	-3.9	< .0001	***
PERSPECTIVE	RAYCASTING	-1.0	.34	

Table 8: *Path Deviation* for linear tracing

(a) TECH ($F_{2,22} = 39.53$, $p < .0001$, $\eta^2_G = 0.56$)

comparisons		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-4.1	< .0001	***
EVERYWHERE	RAYCASTING	-5.4	< .0001	***
PERSPECTIVE	RAYCASTING	-1.3	.029	*

(b) TECH × SURFACE ($F_{6,66} = 1.89$, $p = .09$, $\eta^2_G = 0.05$)

n/a

A.4 Circular Tracing

Table 9: *Tracing Time* for circular tracing

(a) TECH ($F_{2,22} = 1.32$, $p = .29$, $\eta^2_G = 0.01$)

comparisons		*diff (ms)*	*p-value*	
n/a				

(b) TECH × SURFACE ($F_{6,66} = 6.62$, $p < .0001$, $\eta^2_G = 0.05$)

comparisons for NORMAL		*diff (ms)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-557	1.00	
EVERYWHERE	RAYCASTING	-486	1.00	
PERSPECTIVE	RAYCASTING	71	1.00	
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	206	.79	*
EVERYWHERE	RAYCASTING	1177	.24	
PERSPECTIVE	RAYCASTING	971	.27	
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	1647	.13	
EVERYWHERE	RAYCASTING	2531	.002	**
PERSPECTIVE	RAYCASTING	885	.13	
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-718	.67	
EVERYWHERE	RAYCASTING	-1032	.67	
PERSPECTIVE	RAYCASTING	-314	.91	

Table 10: *Tracing Error* for circular tracing

(a) TECH ($F_{2,22} = 51.32$, $p < .0001$, $\eta^2_G = 0.33$)

comparisons		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-1.4	< .0001	***
EVERYWHERE	RAYCASTING	-2.1	< .0001	***
PERSPECTIVE	RAYCASTING	-0.7	.024	*

(b) TECH × SURFACE ($F_{6,66} = 3.08$, $p < .01$, $\eta^2_G = 0.09$)

comparisons for NORMAL		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.1	.67	
EVERYWHERE	RAYCASTING	-0.9	.0030	**
PERSPECTIVE	RAYCASTING	-0.8	.0079	**
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	-2.4	< .0001	***
EVERYWHERE	RAYCASTING	-3.3	< .0001	***
PERSPECTIVE	RAYCASTING	-0.9	.30	
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	-0.9	.29	
EVERYWHERE	RAYCASTING	-0.3	.72	
PERSPECTIVE	RAYCASTING	-0.6	.72	
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-2.2	.0014	**
EVERYWHERE	RAYCASTING	-4.0	< .0001	***
PERSPECTIVE	RAYCASTING	-1.8	.031	*

Table 11: *Path Deviation* for circular tracing

(a) TECH ($F_{2,22} = 76.11$, $p < .0001$, $\eta^2_G = 0.45$)

comparisons		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-3.0	< .0001	***
EVERYWHERE	RAYCASTING	-2.9	< .0001	***
PERSPECTIVE	RAYCASTING	-0.1	.17	

(b) TECH × SURFACE ($F_{6,66} = 6.72$, $p < .0001$, $\eta^2_G = 0.17$)

comparisons for NORMAL		*diff (mm)*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.0	.78	
EVERYWHERE	RAYCASTING	-1.0	.0014	**
PERSPECTIVE	RAYCASTING	-0.9	.0023	**
comparisons for OBLIQUE				
EVERYWHERE	PERSPECTIVE	-3.4	< .0001	***
EVERYWHERE	RAYCASTING	-3.3	< .0001	***
PERSPECTIVE	RAYCASTING	0.0	.85	
comparisons for IRREGULAR				
EVERYWHERE	PERSPECTIVE	-1.7	.18	
EVERYWHERE	RAYCASTING	-0.5	.59	
PERSPECTIVE	RAYCASTING	1.2	.59	
comparisons for CURVED				
EVERYWHERE	PERSPECTIVE	-6.8	< .0001	***
EVERYWHERE	RAYCASTING	-6.8	< .0001	***
PERSPECTIVE	RAYCASTING	-0.0	.46	

A.5 Subjective Ratings

Table 12: Subjective ratings by TECHNIQUE for short-distance selections

(a) *Satisfaction* ($F_{2,22} = 21.58$, $p < .0001$, $\eta^2_G = 0.66$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.3	.48	
EVERYWHERE	RAYCASTING	2.4	< .0001	***
PERSPECTIVE	RAYCASTING	2.8	< .0001	***

(b) *Intuitiveness* ($F_{2,22} = 3.20$, $p = .060$, $\eta^2_G = 0.23$)

comparisons		*diff*	*p-value*	
n/a				

(c) *Physical Fatigue* ($F_{2,22} = 56.14$, $p < .0001$, $\eta^2_G = 0.84$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.7	.12	
EVERYWHERE	RAYCASTING	-3.7	< .0001	***
PERSPECTIVE	RAYCASTING	-3.0	< .0001	***

Table 13: Subjective ratings by TECHNIQUE for long-distance selections

(a) *Satisfaction* ($F_{2,22} = 7.86$, $p < .0005$, $\eta^2_G = 0.42$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.3	.48	
EVERYWHERE	RAYCASTING	1.3	.012	**
PERSPECTIVE	RAYCASTING	1.7	.0035	***

(b) *Intuitiveness* ($F_{2,22} = 5.97$, $p < .001$, $\eta^2_G = 0.35$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	-1.0	.0083	**
EVERYWHERE	RAYCASTING	-0.8	.056	
PERSPECTIVE	RAYCASTING	0.3	.32	

(c) *Physical Fatigue* ($F_{2,22} = 15.26$, $p < .0001$, $\eta^2_G = 0.58$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.2	.52	
EVERYWHERE	RAYCASTING	-3.4	< .0001	***
PERSPECTIVE	RAYCASTING	-3.3	< .0001	***

Table 14: Subjective ratings by TECHNIQUE for tracing

(a) *Satisfaction* ($F_{2,22} = 15.26$, $p < .0001$, $\eta^2_G = 0.58$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	0.1	.94	
EVERYWHERE	RAYCASTING	2.3	.0002	***
PERSPECTIVE	RAYCASTING	2.3	.0002	***

(b) *Intuitiveness* ($F_{2,22} = 1.06$, $p = .36$, $\eta^2_G = 0.09$)

comparisons		*diff*	*p-value*	
n/a				

(c) *Physical Fatigue* ($F_{2,22} = 60.59$, $p < .0001$, $\eta^2_G = 0.85$)

comparisons		*diff*	*p-value*	
EVERYWHERE	PERSPECTIVE	-0.3	.049	*
EVERYWHERE	RAYCASTING	-3.9	< .0001	***
PERSPECTIVE	RAYCASTING	-3.6	< .0001	***